ADDING MODEL UNCERTAINTY TO DEPTH PREDICTION

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Eric Wu

August 2019

ABSTRACT

Disparity and depth estimation of images is a fundamental problem for computer vision. Recent work has shown that convolutional neural networks are effective at both monocular and binocular depth prediction. However, standard neural networks do not give any information about the confidence of their predictions, making it impossible to know if a measurement could be inaccurate. In this work, we add Bayesian uncertainty to pretrained convolutional neural networks. Testing the networks on a synthetic dataset shows that the uncertainty is able to give confidence levels that are linked with the accuracy of the model output. Additionally, masking high uncertainty areas increases the remaining accuracy at the cost of decreasing the completeness of the output.

BIOGRAPHICAL SKETCH

Eric Wu is a second-year Master's student at Cornell University, from which he has already received a bachelor's degree. When he started college, he planned to major in Electrical Engineering, but was waylaid by a class on computer architecture that convinced him to switch to Computer Science. He has enjoyed working on artificial intelligence research since 2016, and is particularly interested in the applications of computer vision.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Professor Kavita Bala. During my years of both undergraduate and graduate studies, she was incredibly dedicated and supportive. Her guidance and advice whenever I encountered problems was invaluable, and I could not have imagined having a better advisor and mentor for my Master's study.

I also would like to thank Professor David Easley for his encouragement and insightful comments. His willingness to give his time so generously to participate in my committee has been very much appreciated.

Finally, I wish to thank my family for their support and encouragement throughout my study and life.

This work was supported by the Office of Naval Research under the PERISCOPE MURI Contract #N00014-17-1-2699. Their support is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The extraction of depth from images has been an essential problem in computer vision. The per-pixel depth of an image is a necessary component in many applications, including autonomous vehicle navigation, 3D reconstruction, robotics, and object recognition. It has been subject to many decades of study that has developed a large and diverse selection of algorithms for depth estimation.

While highly accurate depth information can be obtained from LIDAR and other active sensing techniques, the required equipment can be expensive or otherwise impracticable. Therefore, numerous methods have been established to determine depth solely based on camera data.

Monocular depth estimation uses a single camera. However, a single image provides much less data than multiple images. Monocular depth thus usually utilizes a sequence of images in conjunction with methods such as Structure from Motion (SfM) [26] to derive depth and reconstruct scenes. Stereo depth, as opposed to monocular depth, is inspired by the human binocular vision system and uses two or more cameras. A single pair of images is sufficient to provide depth, although as with monocular depth estimation, a sequence of images will also provide higher accuracy.

Because of the complexity of conventional algorithm-based solutions, convolutional neural networks (CNNs) have been also been used to attempt an artificial intelligence-based solution. CNNs have recently been successful [8, 14, 22, 28] with a myriad of network architectures. They provide significant gains in accuracy and speed compared with conventional approaches.

However, by default, CNNs do not generate model uncertainty. Uncertainty values have been important in many fields of study [7,11]. Some areas are shifting towards preferring Bayesian uncertainty. If given uncertainty values, computers could treat more uncertain inputs differently, such as requiring human intervention in critical applications.

It has been shown that it is possible to add Bayesian uncertainty [5] to classification networks by using extra network layers. This is unlike the predictive category probabilities of the softmax layer at the end of a classification model, which may be wrongly interpreted as a confidence value. It is possible for a model to have a high category probability with low certainty. Meanwhile, adding certain layers is shown to actually represent an estimation of model uncertainty.

This work attempts to add Bayesian uncertainty to convolutional neural networks in order to obtain information about the accuracy of its depth output. The networks are tested on a synthetic dataset against conventional stereo depth methods. It is then shown that model uncertainty can be used to filter network output to obtain considerably more accurate results, at the expense of the completeness of output.

CHAPTER 2

RELATED WORK

Due to its importance, stereo depth has a long history of attempts to increase accuracy with innovative algorithms and novel active measuring techniques. According to Scharstein and Szeliski [20], conventional stereo depth algorithms consist of a subset of four steps: matching cost computation, cost aggregation, disparity computation, and disparity refinement. Depth can then be directly calculated from disparity if camera parameters are known.

2.1 Block-Matching

In general, block-matching algorithms attempt to find matching blocks between two images. The search area is usually limited to a small window around the block, and the size of the blocks is usually adjustable. Each block within the search window is compared to the target block, with some distance metric used to quantitatively determine the best matching block in the area. The disparity (pixel distance) between the best block and the target block is then computed. Figure 2.1 illustrates this process.

The simplest such algorithm for stereo depth is Sum of Squared Differences (SSD) block-matching, which attempts to find disparity by matching blocks between the left and right stereo images. The metric is the sum of squared differences between the pixel values. The block with the smallest sum is used as the match. More complex distance metrics may be used, such as a normalized cross-correlation to mitigate variation with linear changes in brightness, a Hamming

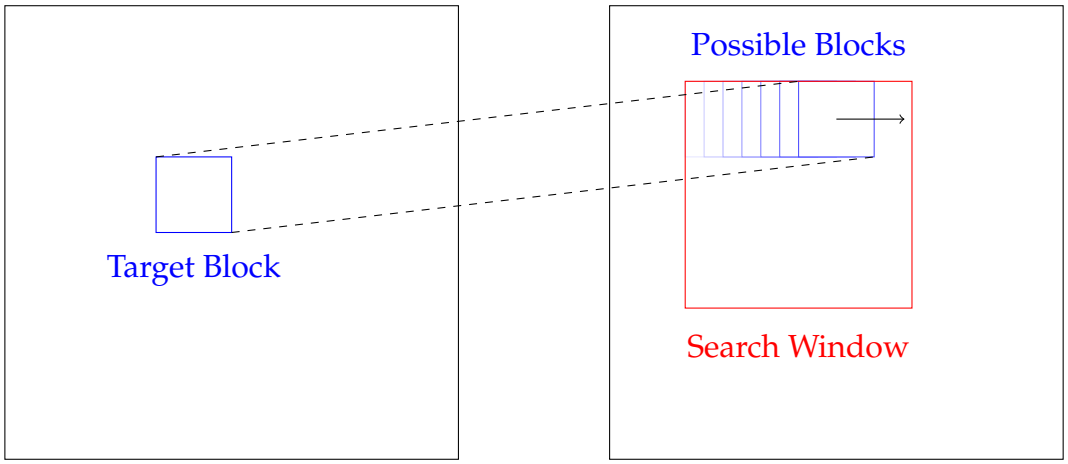


Figure 2.1: Block matching algorithm between two images. The blue square in left image is the target block. The large red square in right image represents the search window, and blue squares are blocks that are checked for matches.

distance between feature descriptors, or a rank transform [20].

OpenCV [1], a real-time computer vision library, includes two block matching algorithms with filtering for use in stereo depth calculation. StereoBM employs naive SSD block matching similar to the above description. The StereoBinarySGBM class instead implements a modified H. Hirschmuller algorithm [12], which is a more complex block matching algorithm. It uses semiglobal matching (SGM), which computes a fast approximation of a smoothness constraint by using pathwise optimizations from 5 directions (unlike 8 directions in the original paper). SGM also calculates for subpixel disparity. Subpixel disparity allows for higher depth accuracy, unlike naive SSD block matching, by discerning disparities that may be shifted only a part of a pixel.

2.2 Structured Light

Structured Light is an active stereo method that typically requires specialized texture projectors. These project some known light pattern onto the scene, allowing cameras to try and extract scene information based on the deformation of the pattern's reflection off objects. The method can be used with only one camera.

However, structured light is sensitive to external interference, such as bright light sources or anything that resembles the texture. Very distant objects can also be difficult to detect due to attenuation of the projected light. Some projectors use invisible wavelengths or extremely high refresh rates with alternating patterns to mitigate external interference. Scharstein and Szeliski [21] were able to remove the need for calibration while yielding a per-pixel disparity map. With a single camera and two light projectors, the disparity for semi-occluded objects was able to be calculated.

There has also been research into unstructured light, which does not require special projectors but instead uses moving illumination such as a person sweeping a flashlight, and temporal stereo, using images from more than one point in time. Zhang *et al.* [29] uses spatial and temporal appearance variation in scenes to achieve higher accuracy. Davis *et al.* [3] presents a method using unstructured light and "hybrid spatiotemporal matching" to achieve better results than spatial-only stereo.

2.3 Artificial Intelligence

Artificial Intelligence provides a variety of methods to approach depth extraction. These include neural networks, which are computing systems that "learn" to perform specific tasks by training on data. Such neural networks usually take images, possibly with filters applied, then run the image through a series of layers. Each layer performs an operation on the pixel values from the previous layer. The final layer produces the desired output.

There has been significant progress recently [2, 8, 13, 14, 17] in increasing the accuracy of neural networks for depth prediction. MegaDepth [13] and MonoDepth [8] are monocular depth neural networks that are able to predict depth given only one observation (image) of a scene. However, they are unable to correctly scale to the exact numerical depth because of the lack of a second image. MegaDepth [13] uses an "hourglass" shaped network, which is a type of encoder-decoder architecture that has also been used in other depth networks such as PSMNet [2].

Other approaches include a hybrid neural network and conventional method [28] that post-processes the network output. To match over multiple (> 2) images, an n-way network [9] has been proposed. Luo *et al.* [15] treat the problem as a multi-class classification instead of the typical regression setup that is used when the output is a continuous numerical range.

CHAPTER 3

APPLYING UNCERTAINTY

Most methods of depth estimation use disparity as an intermediate step. In this section, we will show how uncertainty will be added to various methods, starting from disparity. We will use the definition of disparity as the horizontal displacement between a pair of corresponding pixels from the right to the left image. If a point (x, y) in the left image is found to be located at $(x - d, y)$ in the right image, then the disparity is d. For a parallel pair of identical cameras at the same height, the relation of disparity to depth is

$$Z = \frac{fB}{d} \tag{3.1}$$

where Z is the depth in meters, f is the focal length in pixels, B is the baseline (distance between the two cameras) in meters, and d is the disparity in meters. If the cameras are not parallel and at the same height, then it would be necessary to rectify the images before performing any operation.

3.1 Block Matching

We will use block matching methods as a baseline. Since there is no "uncertainty" built-in to block matching, the expected camera calibration error is used instead. This value is the RMS pixel error between the observed and actual locations of calibration points, and depends on the camera used. As disparity is defined as a pixel displacement, the camera calibration error can also be used as the expected disparity error.

Then, the expected depth error from 3.1 would be

$$\Delta Z = |Z - Z'|$$

where Z is the calculated depth and Z' is the depth at expected disparity error. Calculating ΔZ results in

$$\Delta Z = \left| \frac{fB}{d} - \frac{fB}{d+e} \right| \tag{3.2}$$

where e is the expected disparity or calibration error. The error for block matching is only relative to the depth and the camera parameters.

3.1.1 Dropout Approximation

Dropout [23] is a technique for addressing overfitting in neural networks. Overfitting is a modeling error where the network can accurately predict on the training dataset but cannot predict unseen data, and is highly undesirable. Dropout is presented as a layer that, during training, drops out neural network units with a set probability p. At test or prediction time, dropout attempts to average all possible predictions from the entire network. Since it is not feasible to explicitly run through all permutations of the model, it instead calculates the expected output of each network unit by scaling node weights.

It is possible to approximate Bayesian uncertainty to any neural network by utilizing Dropout [5]. If a dropout layer is applied before every weight (convolution) layer, the layers become a mathematical approximation of a deep Gaussian process. Then, performing T stochastic forward passes through the network makes a Monte Carlo estimate of the predictive distribution called *MC dropout*, from which the model's predictive mean and predictive uncertainty can be estimated. The dropout layers are kept active even in test/prediction mode,

and each batch can be run through the network multiple times either serially or concurrently to speed up runtime.

To evaluate uncertainty on MegaDepth and PSMNet, we add a dropout layer before every convolution layer with a dropout probability of $p = 0.2$. The default probability of $p = 0.5$ was found to need a large number of forward passes T before convergence to a sensible approximation, and testing revealed that 0.2 did not need as many passes ($T = 25$) while also providing a reasonable approximation.

3.2 MegaDepth and PSMNet

We will use the above to add uncertainty to MegaDepth [13] and PSMNet [2]. These are both neural networks that use an "hourglass" structure [16] that has achieved state-of-the-art results on human pose estimation.

MegaDepth [13] is a monocular depth neural network that predicts depth given only one observation (image) of a scene. It does not use disparity as an intermediate step, so its model uncertainty can be used directly as depth uncertainty. However, because it only uses one image, it is unable to scale its output to the correct real depth. Therefore, error calculations for MegaDepth will use the log-space domain unless otherwise noted.

PSMNet [2] is a stereo depth neural network that takes a left and right image as input, and outputs subpixel disparity for the left images. The network uses a pyramid pooling [10] section that feeds into either a basic sequential or a stacked hourglass architecture. Pyramid pooling has been used successfully

in other optical networks such as SPyNet [19]. The pyramid pooling section incorporates hierarchical context information in the image, while the hourglass generates three main disparity maps that are all used in training. However, only the last map is used as the final disparity for prediction, and Equation 3.1 can be used to calculate actual depth.

3.2.1 Converting to Depth Uncertainty

The uncertainty generated above is related to the disparity for most stereo depth networks, not the depth. In order to convert it to uncertainty in depth, propagation of the uncertainty is required. Depth can be calculated from disparity if the camera parameters and relative positions are known according to Equation 3.1. Unlike the calculation of maximum error for block matching in Equation 3.2, it is necessary to use the propagation of uncertainty for a function in the form of $f = \frac{A}{B}$. The propagation for such functions is defined as

$$\sigma_f \approx |f| \sqrt{\left(\frac{\sigma_A}{A}\right)^2 + \left(\frac{\sigma_B}{B}\right)^2 + 2\frac{\sigma_{AB}}{AB}} \tag{3.3}$$

where σ_A and σ_B are the standard deviations of A and B, σ_{AB} is the covariance, and σ_f is function's standard deviation. Since fB depends only on fixed camera parameters, it is constant and the standard deviation and covariance are $\sigma_A = \sigma_{AB} = 0$. The equation then becomes

$$\sigma_f \approx \left| f\frac{\sigma_B}{B} \right|$$

or, replacing the function variables with the appropriate names,

$$\sigma_Z \approx \left| Z\frac{\sigma_d}{d} \right| \tag{3.4}$$

By applying this equation to the disparity, disparity uncertainty, and calculated depth, we can calculate the relevant depth uncertainty when necessary.

CHAPTER 4

EXPERIMENTS AND RESULTS

For all following sections, we used the stacked hourglass option of PSMNet. After dropout was added, the modified MegaDepth and PSMNet networks were fine tuned starting from pretrained weights provided by the authors of the networks. MegaDepth's weights were trained on the MegaDepth's included dataset. PSMNet's weights were trained on the KITTI 2012 dataset [6, 25]. The dataset used for this work for tuning and testing was the ECCV 2018 3D Reconstruction Challenge dataset [24]. It includes ten thousand synthetically generated stereo image pairs of a garden. The image pairs replicate a robot with multiple stereo cameras in different poses traversing through the garden.

Both PSMNet and MegaDepth were implemented in PyTorch [18]. PSMNet used the Adam optimizer with ($\beta_1 = 0.9, \beta_2 = 0.999$). During training, the input was randomly cropped to a height of 256 and width of 512. The maximum disparity was set to 192. The network was fine tuned for 50 epochs at an initial learning rate of 0.001 for 10 epochs and then a learning rate of 0.0001. The batch size was set to 2 for training and 1 for testing. The fine tuning process took approximately 12 hours on a Tesla K40 GPU. A training/validation split of 80%/20% was used.

MegaDepth was tuned with SGD and an initial learning rate of 0.0001. The learning rate was reduced automatically on loss plateau. The network was also fine tuned for 50 epochs with the batch size was set to 1 for both training and testing. The fine tuning process took approximately 10 hours, with the same training/validation split of 80%/20%.

4.1 Baselines

As baseline comparison, we used OpenCV's StereoBM and StereoBinarySGBM block matching methods [1] with a camera disparity error of 0.10. The default block size of 21 was used for StereoBM method, while a block size of 5 was used for StereoBinarySGBM. A maximum disparity of 112 was used for both. StereoBinarySGBM also used the recommended P1 and P2 parameters, a maximum value of 10 pixels in the left-right disparity check, a uniqueness ratio of 10, a speckle window size of 100, and a maximum disparity variation of 1. These parameters were all chosen to be within OpenCV recommended values.

4.2 Results

For all depth predictions, any sky or invalid pixels that were represented with zero disparity in the ground truth were masked out in error calculation, since these would essentially have infinite depth. This could also have been replicated with a semantic network that masks out sky pixels.

Figure 4.1 illustrates a sample image and Figure 4.2 shows the outputs of depth predictions. For the baseline methods, uncertainty is only weakly related with the error. For the networks, areas with high error also have high uncertainty, especially in ill-posed regions such as thin objects or near edges. Among all methods, PSMNet seems to most correctly label higher error areas with higerh uncertainty.

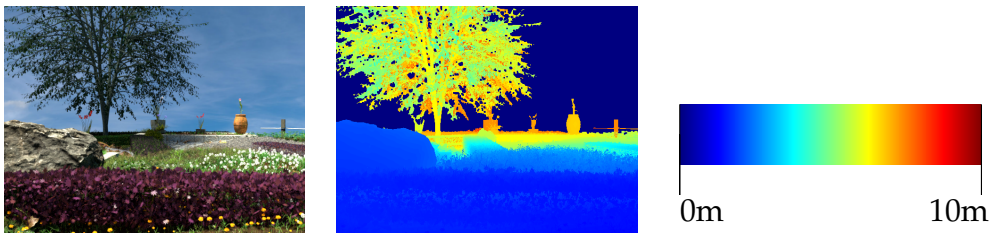
4.3 Uncertainty Filtering

A threshold δ can be applied to the uncertainty in the network models. By varying the threshold and filtering pixels where the uncertainty $> \delta$, areas of higher error pixels can be masked out. We use a threshold $\delta = 3$ for disparity uncertainty and mask out such pixels for the methods which utilize disparity. Results are shown in Table 4.1. The D1 metric from the KITTI 2015 competition [6,25] is used as the accuracy metric, which is defined as the percentage of disparity values that are not within ≤ 3 pixels or $\leq 5\%$ of the ground truth disparity. Since the block matching methods have a fixed disparity error, filtering does not change the result. For PSMNet, there is a 0.38% reduction in its D1 score, which is a noteworthy improvement. MegaDepth does not give any disparity.

Table 4.1: Disparity D1 metric for predictions.

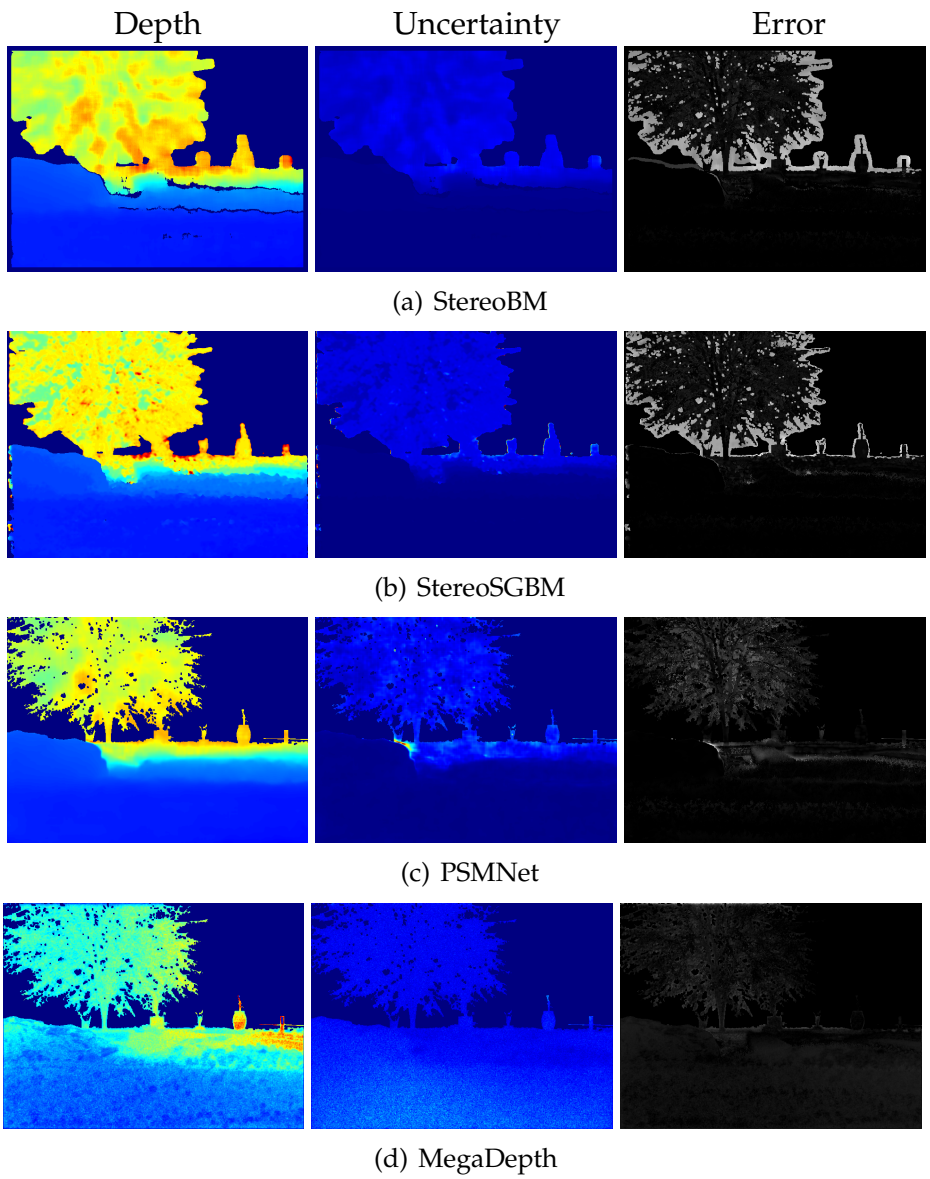
Method	No Filtering	Filtering
StereoBM	24.78%	24.78%
StereoBinarySGBM	6.77%	6.77%
PSMNet	2.57%	2.19%

Depth uncertainty was calculated using Equation 3.3 where necessary. We



(a) Left Image (b) Ground Truth Depth (c) Depth Color Key

Figure 4.1: Sample test image and ground truth depth. A color key for the depth maps is also provided.



(a) StereoBM

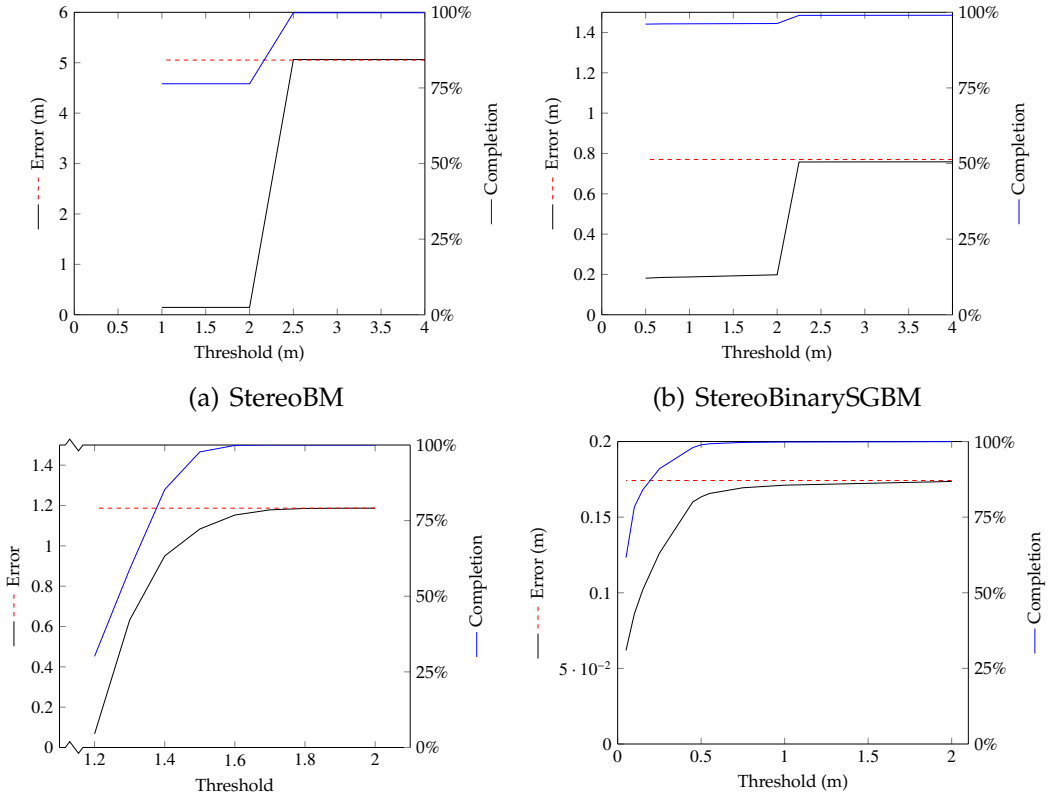
(b) StereoSGBM

(c) PSMNet

(d) MegaDepth

Figure 4.2: Comparison of depth estimations by method. Uncertainty values are multiplied by 5 for visualization purposes. The error images display the absolute value of the difference between the ground truth and the prediction. MegaDepth's outputs are not to scale.

apply the same threshold filtering to MegaDepth and PSMNet at varying thresholds. Figure 4.3 shows plots of threshold vs. average error for baselines along with MegaDepth and PSMNet, using L1 error (average absolute value of the difference to ground truth). For both networks, reducing the threshold re-



(a) StereoBM (b) StereoBinarySGBM

(c) MegaDepth (d) PSMNet

Figure 4.3: Mean depth error per pixel at varying uncertainty thresholds. Black curve uses thresholding, red line is error without thresholding. Blue curve is completion percentage.

sults in lower remaining error, and a higher accuracy at the expense of masking out more portions of the image. The threshold could be adjusted according to application needs in order to balance accuracy and completeness.

4.4 Limitations

Running all 2,000 validation images took over 5 hours per network, with $T = 25$ forward passes per image. This is not an insignificant amount of time, and it would be infeasible to run uncertainty calculations for real time video.

CHAPTER 5

CONCLUSION

It has been presented that the addition of uncertainty to neural networks is useful in filtering out inaccurate results. The modification is relatively simple and is possible for any neural network, but slows prediction speed. Experiments on a synthetic dataset show that by masking regions with an adjustable uncertainty threshold, the remaining regions had considerably higher accuracy. The model uncertainty can be used to obtain more accurate results, at the expense of the completeness of output.

5.1 Future Work

For future work, it would be necessary to test on a non-synthetic dataset. Synthetic data makes it easy to generate images at arbitrary locations in the environment while using uniform camera parameters. However, it does not accurately represent challenges that would be encountered in a real dataset, such as noise, artifacts, obstructions, and reflections.

A limitation of the current method is the speed of the uncertainty determination. With the tested parameters, 25 forward passes per image were required for a reasonable convergence. Normally, a single pass suffices to gain depth information on most state-of-the-art networks. Modeling uncertainty takes a significantly higher amount of time than one pass. The time needed could be reduced by running in parallel on multiple GPUs, but this would increase application cost. It may also be possible to do a faster approximation of the forward

passes at the cost of uncertainty accuracy.

Finally, it may be interesting to modify the network to incorporate other data. Semantic information has been shown to be useful in augmenting stereo depth networks [4, 27], improving results in ambiguous areas. Hybrid conventional and AI approaches or temporal and spatial approaches may also be possible. Combining such additional information with uncertainty could improve accuracy. Uncertain areas may also help determine where more information about a scene is needed.

BIBLIOGRAPHY

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[2] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018.

[3] James Davis, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–359. IEEE, 2003.

[4] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[5] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452, 2015.

[8] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[9] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1586–1594, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[11] Stefan Herzog and Dirk Ostwald. Experimental biology: sometimes bayesian statistics are better. *Nature*, 494(7435):35, 2013.

[12] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007.

[13] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[14] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.

[15] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016.

[16] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[17] Jiahao Pang, Wenxiu Sun, Jimmy SJ Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 887–895, 2017.

[18] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[19] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4161–4170, 2017.

[20] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[21] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps

using structured light. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003.

[22] Amit Shaked and Lior Wolf. Improved stereo matching with constant highway networks and reflective confidence learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4641–4650, 2017.

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[24] Radim Tylecek, Torsten Sattler, Hoang-An Le, Thomas Brox, Marc Pollefeys, Robert B. Fisher, and Theo Gevers. 3d reconstruction meets semantics: Challenge results discussion. Technical report, ECCV Workshops, September 2018.

[25] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.

[26] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979.

[27] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 636–651, 2018.

[28] Jure Zbontar, Yann LeCun, et al. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016.

[29] Li Zhang, Brian Curless, and Steven M Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–367. IEEE, 2003.